SOME LANGUAGES RECOGNIZED BY TWO-WAY FINITE AUTOMATA WITH QUANTUM AND CLASSICAL STATES *

Shenggen Zheng[1,†] Daowen Qiu[1,2,3,‡] Lvzhou Li[1, §]

[1] *Department of Computer Science, Sun Yat-sen University, Guangzhou 510006, China*

[2] *SQIG–Instituto de Telecomunicações, Departamento de Matemática, Instituto Superior Técnico, TULisbon, Av. Rovisco Pais 1049-001, Lisbon, Portugal*

[3] *The State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100080, China*

Abstract

Two-way finite automata with quantum and classical states (2QCFA) were introduced by Ambainis and Watrous, and it was shown that 2QCFA have superiority over *two-way probabilistic finite automata* (2PFA) for recognizing some non-regular languages such as the language $L_{eq} = \{a^n b^n \mid n \in \mathbf{N}\}$ and the palindrome language $L_{pal} = \{\omega \in \{a,b\}^* \mid \omega = \omega^R\}$, where x^R is x in the reverse order. It is interesting to find more languages like these that witness the superiority of 2QCFA over 2PFA. In this paper, we consider the language $L_m = \{xcy \mid \Sigma = \{a,b,c\}, x,y \in \{a,b\}^*, c \in \Sigma, |x| = |y|\}$ that is similar to the middle language $L_{middle} = \{xay \mid x,y \in \Sigma^*, a \in \Sigma, |x| = |y|\}$. We prove that the language L_m can be recognized by 2QCFA with one-sided error in polynomial expected time. Also, we show that L_m can be recognized by 2PFA with bounded error, but only in exponential expected time. Thus L_m is another witness of the fact that 2QCFA are more powerful than their classical counterparts.

Keywords: Computing models; Probabilistic finite automata; Quantum finite automata.

*This work is supported in part by the National Natural Science Foundation (Nos. 60873055, 61073054, 61100001), the Natural Science Foundation of Guangdong Province of China (No. 10251027501000004), the Fundamental Research Funds for the Central Universities (Nos. 10lgzd12,11lgpy36), the Research Foundation for the Doctoral Program of Higher School of Ministry of Education (Nos. 20100171110042, 20100171120051) of China, the China Postdoctoral Science Foundation project (Nos. 20090460808, 201003375), and the project of SQIG at IT, funded by FCT and EU FEDER projects projects QSec PTDC/EIA/67661/2006, AMDSC UTAustin/MAT/0057/2008, NoE Euro-NF, and IT Project QuantTel.

†*E-mail address:* zhengshenggen@gmail.com

‡Corresponding author. *E-mail address:* issqdw@mail.sysu.edu.cn (D. Qiu)

§*E-mail address:* lilvzhou@gmail.com

1 Introduction

Interest in quantum computation has steadily increased since Shor's quantum algorithm for factoring integers in polynomial time [23] and Grover's algorithm of searching in database of size n with only $O(\sqrt{n})$ accesses [10]. Clarifying the power of some fundamental models of quantum computation has attracted wide attentions in the academic community [11, 18]. As we know, algorithms based on *quantum Turing machines* are complicated to implement using today's experiment technology. Therefore, it is natural to consider much more restricted quantum computing models.

As one of the simplest computing models, *deterministic finite automata* (DFA) and *probabilistic finite automata (PFA)* have been deeply studied [12, 19]. Correspondingly, it may be interesting to consider restricted *quantum Turing machines*, such as *quantum finite automata* (QFA). QFA can be thought of as a theoretical model of quantum computers in which the memory is finite and described by a finite-dimensional state space [1], as *finite automata* (FA) are a natural model for classical computing with finite memory [12]. QFA were first introduced independently by Kondacs and Watrous [13], as well as Moore and Crutchfield [17]. As a quantum variant of FA, QFA have attracted wide attentions in the academic community [1, 15, 16, 22, 24, 26]. There are many kinds of QFA having been proposed and studied (e.g., see [21]). The study of QFA is mainly divided into two kinds: one is *one-way quantum finite automata* (1QFA) whose tape heads move one square right at each evolution, and the other *two-way quantum finite automata* (2QFA), in which the tape heads are allowed to move towards right or left, or to be stationary.

Furthermore, according to the measurement times in a computation, 1QFA have two fashions: measure-once 1QFA (MO-1QFA) proposed by Moore and Crutchfield [17] and measure-many 1QFA (MM-1QFA) studied first by Kondacs and Watrous [13]. MO-1QFA are strictly less powerful than their classical counterparts DFA [13, 17], since they recognize only a proper subset of *regular languages* (RL). Though MM-1QFA are more powerful than MO-1QFA [2], they still recognize with bounded error only a proper subset of RL [4].

2QFA, however, are more powerful than their classical counterparts. 2QFA can not only recognize all regular languages, but also recognize the non-regular language $L_{eq} = \{a^n b^n \mid n \in \mathbf{N}\}$ with bounded error in linear time. Note that *two-way deterministic finite automata* (2DFA) recognize the same family of languages as DFA [12], and a 2PFA requires exponential expected time to recognize L_{eq} [8, 9]. Therefore, 2QFA are more powerful than their classical counterparts. However, 2QFA have a disadvantage in the sense that we need at least $\mathbf{O}(\log n)$ qubits to store the positions of the tape head, which is relative to the length of the input.

In order to conquer the above disadvantage, Ambainis and Watrous [3] proposed a different two-way quantum computing model — *two-way finite automata with quantum and classical*

states (2QCFA) in 2002. As an intermediate model between 1QFA and 2QFA, 2QCFA are still more powerful than their classical counterparts. A 2QCFA is essentially a classical 2DFA augmented with a quantum component of constant size, where the dimension of the associated Hilbert space does not depend on the length of the input. In spite of the existing restriction, 2QCFA are more powerful than 2PFA. Indeed, 2QCFA can recognize all regular languages with certainty, and particularly, Ambainis and Watrous [3] proved that this model can also recognize L_{eq} with one-sided error in polynomial expected time and can recognize palindromes $L_{pal} = \{\omega \in \{a,b\}^* \mid \omega = \omega^R\}$ with one-sided error in exponential expected time. Note that no 2PFA can recognize L_{pal} with bounded error.

Several open problems were proposed by Ambainis and Watrous [3], including the problem whether $L_{middle} = \{xay \mid x,y \in \Sigma^*, a \in \Sigma, |x| = |y|\}$ can be recognized by 2QCFA or not. In this paper, we does not aim to answer the above question, but we consider a similar language $L_m = \{xcy \mid \Sigma = \{a,b,c\}, x,y \in \{a,b\}^*, c \in \Sigma, |x| = |y|\}$. We prove that L_m can be recognized by 2QCFA with one-sided error in polynomial expected time. Meanwhile we show that L_m can also be recognized by 2PFA with bounded error, but in exponential expected time. Thus L_m is another witness of the fact that 2QCFA are more powerful than their classical counterparts.

The remainder of this paper is organized as follows. Some computing models and related definitions are introduced in Section 2. In section 3 we describe a 2QCFA for recognizing L_m with one-sided error in polynomial expected time. In section 4 we show L_m can be recognized by 2PFA with bounded error in exponential expected time. Finally, some concluding remarks are made in Section 5.

2 Definitions

We recall the definitions of 2PFA and 2QCFA in this section.

2.1 Definition of two-way probabilistic finite automata

The notation of 2PFA was introduced by Kuklin [14], and then studied by Freivalds [8] and Dwork etc [5, 6, 9].

A 2PFA $\mathcal{M}$ is defined by a 6-tuple

$$\mathcal{M} = (S, \Sigma, \delta, s_0, S_{acc}, S_{rej}) \tag{1}$$

where,

- S is a finite set of classical states;

- Σ is a finite set of input symbols; the tape symbol set $\Gamma = \Sigma \cup \{ \not\vdash, \$\}$, where $\not\vdash \notin \Sigma$ is called the left end-marker and $\$ \notin \Sigma$ is called the right end-marker;
- $s_0 \in S$ is the initial state of the machine;
- $S_{acc} \subset S$ and $S_{rej} \subset S$ are the sets of accepting states and rejecting states, respectively.
- δ is the transition function:

$$(S \setminus (S_{acc} \cup S_{rej})) \times \Gamma \times S \times \{-1, 0, 1\} \to \{0, 1/2, 1\} \tag{2}$$

Essentially, for each state $s \in S$ and each $\sigma \in \Sigma \cup \{ \not\vdash, \$\}$, $\delta(s, \sigma)$ is a coin-tossing distribution[1] on $S \times \{-1, 0, 1\}$, where $d = -1$ means that the tape head moves one square left, $d = 0$ means that the tape head keeps stationary, and $d = 1$ means that the tape head moves one square right. We assume that δ is well defined so that when the tape head is positioned on the left end-marker $\not\vdash$ (right end-marker \$), the tape head will not move left (right) in next step.

The computation of a 2PFA $\mathcal{M}$ on input $\omega \in \Sigma^*$ begins with the initial state s_0 and with the word $\not\vdash\omega\$$ written on the tape where the tape head is positioned on the left end-marker $\not\vdash$. The computation is then governed (probabilistically) by the transition functions δ until $\mathcal{M}$ either accepts ω by entering an accepting state $s_a \in S_{acc}$ or rejects ω by entering a rejecting state $s_r \in S_{rej}$. $\mathcal{M}$ halts when it enters an accepting state or a rejecting state. It should be pointed out that the computation could be infinite if neither an accepting state nor a rejecting state is entered. Let $L \subset \Sigma^*$ and $0 \leq \epsilon < 1/2$. Then a 2PFA $\mathcal{M}$ recognizes L with bounded error if

1. $\forall \omega \in L$, $Pr[\mathcal{M}\ accepts\ \omega] \geq 1 - \epsilon$, and
2. $\forall \omega \notin L$, $Pr[\mathcal{M}\ rejects\ \omega] \geq 1 - \epsilon$.

2.2 Definition of two-way finite automata with quantum and classical states

2QCFA were introduced by Ambainis and Watrous [3] in 2002, and then studied by in [20, 25, 27].

Informally, we describe a 2QCFA as a 2DFA which has access to a constant size of quantum register, upon which it performs quantum transformations and measurements. We would refer the readers to [11, 18] for a detailed overview of quantum computing.

[1] A coin-tossing distribution on finite set Q is a mapping ϕ from Q to {0, 1/2, 1} such that $\sum_{q \in Q} \phi(q) = 1$, which means choosing q with probability $\phi(q)$.

A 2QCFA is specified by a 9-tuple

$$\mathcal{M} = (Q, S, \Sigma, \Theta, \delta, q_0, s_0, S_{acc}, S_{rej}) \tag{3}$$

where,

- Q is a finite set of quantum states;
- S is a finite set of classical states;
- Σ is a finite set of input symbols; the tape symbol set $\Gamma = \Sigma \cup \{ \not\vdash, \$\}$, where $\not\vdash \notin \Sigma$ is called the left end-marker and $\$ \notin \Sigma$ is called the right end-marker;
- $q_0 \in Q$ is the initial quantum state;
- $s_0 \in S$ is the initial classical state;
- $S_{acc} \subset S$ and $S_{rej} \subset S$ are the sets of classical accepting states and rejecting states, respectively.
- Θ is the transition function of quantum states:

$$S \setminus (S_{acc} \cup S_{rej}) \times \Gamma \to \mathcal{U}(\mathcal{H}(Q)) \cup \mathcal{M}(\mathcal{H}(Q)), \tag{4}$$

  where $\mathcal{U}(\mathcal{H}(Q))$ and $\mathcal{M}(\mathcal{H}(Q))$ respectively denote the sets of unitary operators and projective measurements over $\mathcal{H}(Q)$, and $\mathcal{H}(Q)$ represents the Hilbert space with the corresponding base identified with set Q. Thus, $\Theta(s, \gamma)$ corresponds to either a unitary transformation or a projective measurement.

- δ is the transition function of classical states. If $\Theta(s, \gamma) \in \mathcal{U}(\mathcal{H}(Q))$, then δ is

$$S \setminus (S_{acc} \cup S_{rej}) \times \Gamma \to S \times \{-1, 0, 1\}, \tag{5}$$

  which is similar to the transition function defined for 2PFA except that the transition function here is deterministic. If $\Theta(s, \gamma) \in \mathcal{M}(\mathcal{H}(Q))$ which is a projective measurement, assume that the projective measurement with the set of possible eigenvalues $R = \{r_1, \cdots, r_n\}$ and the projector set $\{P(r_i) : i = 1, \cdots, n\}$ where $P(r_i)$ denotes the projector onto the eigenspace corresponding to r_i, the measurement result set will be $R = \{r_1, r_2, \ldots, r_n\}$. Then δ is

$$S \setminus (S_{acc} \cup S_{rej}) \times \Gamma \times R \to S \times \{-1, 0, 1\}, \tag{6}$$

  where $\delta(s, \gamma)(r_i) = (s', d)$ means that when the projective measurement result is r_i, the classical state $s \in S$ scanning $\gamma \in \Gamma$ is changed to state s', and the movement of the tape head is decided by d.

Given an input ω, a 2QCFA $\mathcal{M} = (Q, S, \Sigma, \Theta, \delta, q_0, s_0, S_{acc}, S_{rej})$ proceeds as follows:

At the beginning, the tape head is positioned on $\not\vdash$, the quantum initial state is $|q_0\rangle$, the classical initial state is s_0, and $|q_0\rangle$ will be changed according to $\Theta(s_0, \not\vdash)$.

a. If $\Theta(s_0, \not\vdash) = U \in \mathcal{U}(\mathcal{H}(Q))$, then the quantum state evolves as $|q_0\rangle \to U|q_0\rangle$, and meanwhile, the classical state s_0 is changed to s' according to $\delta(s_0, \not\vdash_1) = (s', d)$. The movement of the tape head is decided by d.

b. If $\Theta(s_0, \not\vdash_1) = M \in \mathcal{M}(\mathcal{H}(Q))$, then the measurement M is performed on $|q_0\rangle$. Let $M = \{P_1, \cdots, P_m\}$ with result set $R = \{r_i\}_{i=1}^m$. After the measurement M has been performed, we get a result $r_i \in R$ with probability $p_i = \langle q_0|P_i|q_0\rangle$, and the quantum state $|q_0\rangle$ changes to $P_i|q_0\rangle/\sqrt{p_i}$. Meanwhile, the classical state changes according to $\delta(s_0, \not\vdash)(r_i) = (s_i, d)$. If $s_i \in S_{acc}$ (S_{rej}), $\mathcal{M}$ accepts (rejects) ω and halts; otherwise, the tape head of $\mathcal{M}$ moves according to the direction d, and continues to read the next symbol.

A computation is assumed to halt if and only if an accepting state or a rejecting classical state is entered.

Let $L \subset \Sigma^*$ and $0 \leq \epsilon < 1/2$. A 2QCFA $\mathcal{M}$ recognizes L with one-sided error if

1. $\forall \omega \in L$, $Pr[\mathcal{M}\ accepts\ \omega] = 1$, and
2. $\forall \omega \notin L$, $Pr[\mathcal{M}\ rejects\ \omega] \geq 1 - \epsilon$.

3 A 2QCFA recognizing the language L_m

We prove that $L_m = \{xcy \mid \Sigma = \{a, b, c\}, x, y \in \{a, b\}^*, c \in \Sigma, |x| = |y|\}$ can be recognized by 2QCFA with one-sided error in polynomial expected time in this section.

Theorem 1. *For any $\epsilon > 0$, there is a 2QCFA $\mathcal{M}$ that accepts any $\omega \in L_m = \{xcy \mid \Sigma = \{a, b, c\}, x, y \in \{a, b\}^*, c \in \Sigma, |x| = |y|\}$ with certainty, rejects any $\omega \notin L_m$ with probability at least $1 - \epsilon$ and halts in polynomial expected time.*

Proof. In order to prove Theorem 1, we consider two matrices U_a and U_c defined as follows:

$$U_a = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & \cos\alpha & \sin\alpha \\ 0 & 0 & -\sin\alpha & \cos\alpha \end{pmatrix}, U_c = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \tag{7}$$

where let $\alpha = \sqrt{2}\pi$. Obviously, U_a and U_c are unitary, and it is easy to get that

$$(U_a)^k = \begin{pmatrix} \cos k\alpha & -\sin k\alpha & 0 & 0 \\ \sin k\alpha & \cos k\alpha & 0 & 0 \\ 0 & 0 & \cos k\alpha & \sin k\alpha \\ 0 & 0 & -\sin k\alpha & \cos k\alpha \end{pmatrix}, \tag{8}$$

and

$$(U_a)^l U_c (U_a)^k = \begin{pmatrix} 0 & 0 & \cos(k-l)\alpha & \sin(k-l)\alpha \\ 0 & 0 & -\sin(k-l)\alpha & \cos(k-l)\alpha \\ \cos(k-l)\alpha & -\sin(k-l)\alpha & 0 & 0 \\ \sin(k-l)\alpha & \cos(k-l)\alpha & 0 & 0 \end{pmatrix}. \tag{9}$$

We now describe a 2QCFA $\mathcal{M}$ with 4 quantum states $\{|q_0\rangle, |q_1\rangle, |q_2\rangle, |q_3\rangle\}$, of which $|q_0\rangle$ is the initial state. $\mathcal{M}$ has three unitary operators: U_a, U_b and U_c where U_a and U_c are given in Eq. (7) and $U_b = U_a$. They can also be described as follows:

$U_a\|q_0\rangle = \cos\alpha\|q_0\rangle + \sin\alpha\|q_1\rangle$	$U_b\|q_0\rangle = \cos\alpha\|q_0\rangle + \sin\alpha\|q_1\rangle$	$U_c\|q_0\rangle = \|q_2\rangle$
$U_a\|q_1\rangle = -\sin\alpha\|q_0\rangle + \cos\alpha\|q_1\rangle$	$U_b\|q_1\rangle = -\sin\alpha\|q_0\rangle + \cos\alpha\|q_1\rangle$	$U_c\|q_1\rangle = \|q_3\rangle$
$U_a\|q_2\rangle = \cos\alpha\|q_2\rangle - \sin\alpha\|q_3\rangle$	$U_b\|q_2\rangle = \cos\alpha\|q_2\rangle - \sin\alpha\|q_3\rangle$	$U_c\|q_2\rangle = \|q_0\rangle$
$U_a\|q_3\rangle = \sin\alpha\|q_2\rangle + \cos\alpha\|q_3\rangle$	$U_b\|q_3\rangle = \sin\alpha\|q_2\rangle + \cos\alpha\|q_3\rangle$	$U_c\|q_3\rangle = \|q_1\rangle$

The automaton $\mathcal{M}$ proceeds as follows:

Check whether the input is of the form xcy ($x, y \in \Sigma^*$). If not, reject.
Otherwise, repeat the following ad infinitum:

1. Move the tape head to the first input symbol and set the quantum state to $|q_0\rangle$.
2. While the currently scanned symbol is not \$, do the following:
   (2-1). If the currently scanned symbol is a or b, perform U_a on the quantum state.
   (2-2). If the currently scanned symbol is c, perform U_c on the quantum state.
   (2-3). Move the tape head one square to the right.
3. Measure the quantum state. If the result is not $|q_2\rangle$, reject.
4. Repeat the following subroutine two times:
   (4-1).Move the tape head to the first input symbol.
   (4-2).Move the tape head one square to the right.
   (4-3).While the currently scanned symbol is not ¢ or \$, do the following:
   Simulate a coin flip. If the result is "head", move right. Otherwise, move left.
5. If both times the process ends at the right end-marker \$, do:
   Simulate k coin-flips. If all results are "heads", accept.

Lemma 2. *If the input $\omega = xcy$ satisfies $|x| = n$, $|y| = m$ and $n = m$, then the quantum state of $\mathcal{M}$ will evolve into $|q_2\rangle$ after loop 2 with certainty.*

Proof. According to Eq. (9), the quantum state after loop **2** can be described as follows:

$$|q\rangle = (U_a)^m U_c (U_a)^n |q_0\rangle \tag{10}$$

$$= \begin{pmatrix} 0 & 0 & \cos(n-m)\alpha & \sin(n-m)\alpha \\ 0 & 0 & -\sin(n-m)\alpha & \cos(n-m)\alpha \\ \cos(n-m)\alpha & -\sin(n-m)\alpha & 0 & 0 \\ \sin(n-m)\alpha & \cos(n-m)\alpha & 0 & 0 \end{pmatrix} |q_0\rangle. \tag{11}$$

Because $n = m$, we get

$$|q\rangle = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} |q_0\rangle = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = |q_2\rangle. \tag{12}$$

So the lemma has been proved. □

Lemma 3. *If the input $\omega = xcy$ satisfies $|x| = n$, $|y| = m$ and $n \neq m$, then $\mathcal{M}$ rejects after step **3** with probability at least $1/(2(n-m)^2+1)$.*

Proof. Starting with state $|q_0\rangle$, $\mathcal{M}$ changes its quantum state to $(U_a)^m U_c (U_a)^n |q_0\rangle$ after loop **2** . According to the analysis given above and Eq. (8-9), we get the quantum state

$$|q\rangle = (U_a)^m U_c (U_a)^n |q_0\rangle = \cos((n-m)\alpha)|q_2\rangle + \sin((n-m)\alpha)|q_3\rangle \tag{13}$$

$$= \cos(\sqrt{2}(n-m)\pi)|q_2\rangle + \sin(\sqrt{2}(n-m)\pi)|q_3\rangle. \tag{14}$$

The probability of observing $|q_3\rangle$ is $\sin^2(\sqrt{2}(n-m)\pi)$ in step **3**. Without loss of generality, we assume that $n - m > 0$. Let l be the closest integer to $\sqrt{2}(n-m)$. If $\sqrt{2}(n-m) > l$, then $2(n-m)^2 > l^2$. So we get $2(n-m)^2 - 1 \geq l^2$ and $l \leq \sqrt{2(n-m)^2-1}$. We have

$$\sqrt{2}(n-m) - l \geq \sqrt{2}(n-m) - \sqrt{2(n-m)^2-1} \tag{15}$$

$$= \frac{(\sqrt{2}(n-m) - \sqrt{2(n-m)^2-1})(\sqrt{2}(n-m) + \sqrt{2(n-m)^2-1})}{\sqrt{2}(n-m) + \sqrt{2(n-m)^2-1}} \tag{16}$$

$$= \frac{1}{\sqrt{2}(n-m) + \sqrt{2(n-m)^2-1}} > \frac{1}{2\sqrt{2}(n-m)}. \tag{17}$$

Because l is the closest integer to $\sqrt{2}(n-m)$, we have $0 < \sqrt{2}(n-m) - l < 1/2$. Let $f(x) = sin(x\pi) - 2x$. We have $f''(x) = -\pi^2 \sin(x\pi) \leq 0$ when $x \in [0, 1/2]$. That is to say, $f(x)$ is concave in $[0, 1/2]$, and we have $f(0) = f(1/2) = 0$. So for any $x \in [0, 1/2]$, it holds that $f(x) \geq 0$, that is, $\sin(x\pi) \geq 2x$. Therefore, we have

$$\sin^2(\sqrt{2}(n-m)\pi) = \sin^2((\sqrt{2}(n-m) - l)\pi) \tag{18}$$

$$\geq (2(\sqrt{2}(n-m)-l))^2 = 4(\sqrt{2}(n-m)-l)^2 \tag{19}$$

$$> 4(\frac{1}{2\sqrt{2}(n-m)})^2 = \frac{1}{2(n-m)^2} > \frac{1}{2(n-m)^2+1}. \tag{20}$$

If $\sqrt{2}(n-m) < l$, then $2(n-m)^2 < l^2$. So we get $2(n-m)^2+1 \leq l^2$ and $l \geq \sqrt{2(n-m)^2+1}$. We have

$$\sqrt{2}(n-m) - l \leq \sqrt{2}(n-m) - \sqrt{2(n-m)^2+1} \tag{21}$$

$$= \frac{(\sqrt{2}(n-m) - \sqrt{2(n-m)^2+1})(\sqrt{2}(n-m) + \sqrt{2(n-m)^2+1})}{\sqrt{2}(n-m) + \sqrt{2(n-m)^2+1}} \tag{22}$$

$$= \frac{-1}{\sqrt{2}(n-m) + \sqrt{2(n-m)^2+1}} < \frac{-1}{2\sqrt{2(n-m)^2+1}}. \tag{23}$$

It follows that

$$l - \sqrt{2}(n-m) > \frac{1}{2\sqrt{2(n-m)^2+1}}. \tag{24}$$

Because l is the closest integer to $\sqrt{2}(n-m)$, we have $0 < l - \sqrt{2}(n-m) < 1/2$. Therefore, we have

$$\sin^2(\sqrt{2}(n-m)\pi) = \sin^2((\sqrt{2}(n-m)-l)\pi) \tag{25}$$

$$= \sin^2((l-\sqrt{2}(n-m))\pi) \geq (2(l-\sqrt{2}(n-m)))^2 \tag{26}$$

$$= 4(l-\sqrt{2}(n-m))^2 > 4(\frac{1}{2\sqrt{2(n-m)^2+1}})^2 = \frac{1}{2(n-m)^2+1}. \tag{27}$$

So the lemma has been proved. □

Simulation of a coin flip in loops **4** and **5** is a key component in the above algorithm. We will show that coin-flips can be simulated by 2QCFA.

Lemma 4. *A coin flip can be simulated by 2QCFA $\mathcal{M}$ with a unitary operation and a measurement.*

Proof. A projective measurement $M = \{P_0, P_1\}$ is defined by

$$P_0 = |p_0\rangle\langle p_0|, P_1 = |p_1\rangle\langle p_1|. \tag{28}$$

The results 0 and 1 represent the "tail" and "head" of a coin flip, respectively. A unitary operator U is given by

$$U = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{pmatrix}. \tag{29}$$

The unitary operator U changes the state as follows:

$$|p_0\rangle \to |\psi\rangle = \frac{1}{\sqrt{2}}(|p_0\rangle + |p_1\rangle), \quad |p_1\rangle \to |\phi\rangle = \frac{1}{\sqrt{2}}(|p_0\rangle - |p_1\rangle). \tag{30}$$

Suppose now that the machine starts with the state $|p_0\rangle$, changes its state by U, and then measures the state with M. Then we will get the result 0 or 1 with probability $\frac{1}{2}$. This is similar to a coin flip process. □

Lemma 5. *Every execution of loops* ***4*** *and* ***5*** *leads to acceptance with probability* $1/2^k(n+m+2)^2$.

Proof. Loop **4** is two times of random walk starting at location 1 and ending at location 0 (the left end-marker ¢) or at location $n+m+2$ (the right end-marker \$). It can be known from probability theory that the probability of reaching the location $n+m+2$ is $1/(n+m+2)$ (see Chapter14.2 in [7]). Repeating it twice and flipping k coins, we get the probability $1/2^k(n+m+2)^2$. □

Let $k = 1 + \lceil \log_2 1/\varepsilon \rceil$, then $\varepsilon \geq 1/2^{k-1}$. If $\omega = xcy$ satisfies $|x| = |y| = n$, loop **2** always changes $|q_0\rangle$ to $|q_2\rangle$, and $\mathcal{M}$ never rejects after the measurement in step **3**. After loops **4** and **5**, the probability of $\mathcal{M}$ accepting ω is $1/2^k(2n+2)^2$. Repeating loops **4** and **5** for cn^2 times, the accepting probability is

$$Pr[\mathcal{M} \textit{ accepts } \omega] = 1 - (1 - \frac{1}{2^k(2n+2)^2})^{cn^2}, \tag{31}$$

and this can be made arbitrarily close to 1 by selecting constant c appropriately.

Otherwise, if $\omega = xcy$ satisfies $|x| = n$, $|y| = m$ and $n \neq m$, $\mathcal{M}$ rejects after loop **2** and step **3** with probability

$$P_r > \frac{1}{2(n-m)^2+1} \tag{32}$$

according to Lemma 3. $\mathcal{M}$ accepts after loops **4** and **5** with probability

$$P_a = 1/2^k(n+m+2)^2 \leq \varepsilon/2(n+m+2)^2. \tag{33}$$

If we repeat the whole algorithm indefinitely, the probability of $\mathcal{M}$ rejecting input ω is

$$Pr[\mathcal{M} \textit{ rejects } \omega] = \sum_{i \geq 0} (1-P_a)^i (1-P_r)^i P_r \tag{34}$$

$$= \frac{P_r}{P_a + P_r - P_a P_r} > \frac{P_r}{P_a + P_r} \tag{35}$$

$$> \frac{1/(2(n-m)^2+1)}{\varepsilon/2(n+m+2)^2 + 1/(2(n-m)^2+1)} \tag{36}$$

$$> \frac{1/2}{1/2 + \varepsilon/2} = \frac{1}{1+\varepsilon} > 1 - \varepsilon. \tag{37}$$

If we assume that the input $\omega = xcy$ where $|x| = n$ and $|y| = m$, then loop **1** takes $\mathbf{O}(n+m)$ time at worst cases, loop **2** and step **3** take $\mathbf{O}(n+m)$ time exactly, and loops **4** and **5** take $\mathbf{O}((n+m)^2)$ time. The expected number of repeating the algorithm is $\mathbf{O}((n+m)^2)$. Hence, the expected running time of $\mathcal{M}$ is $\mathbf{O}((n+m)^4)$. □

4 A 2PFA recognizing the language L_m

L_m looks like the language $L_{middle} = \{xay \mid x, y \in \Sigma^*, a \in \Sigma, |x| = |y|\}$ which has been proved to be not recognizable by 2PFA in Dwork and Stockmeyer's paper[6]. However, we will prove that L_m can be recognized by 2PFA in this section, but the expected time needed is exponential.

Theorem 6. *For any $\epsilon > 0$, there is a 2PFA $\mathcal{M}$ that accepts any $\omega \in L_m = \{xcy \mid \Sigma = \{a, b, c\}, x, y \in \{a, b\}^*, c \in \Sigma, |x| = |y|\}$ at least $1 - \epsilon$, rejects any $\omega \notin L_m$ with probability at least $1 - \epsilon$.*

Proof. We assume that the input $\omega \in \{a, b, c\}^*$ has the form $\omega = xcy$ where $|\omega| = l$, $|x| = n$, and $|y| = m$. Let k be a positive integer. The algorithm for a 2PFA to recognize L_m is described as follows:

Checks whether the length of input $l = |\omega|$ is odd. If not, rejects.
Otherwise, repeat the following ad infinitum:
1. Move the tape head to symbol c. If there is not symbol c in ω, rejects.
2. Simulate a coin flip, and do the following:
(2-1). If the outcome is "head", simulate $k(2n + 2)$ coin-flips,
and move the tape head left to keep count.
(2-2). Otherwise, simulate $k(2m + 2)$ coin-flips,
and move its tape head right to keep count.
(2-3). If all $k(2n + 2)$ or $k(2m + 2)$ flips have outcome "heads" in either case,
reject.
3. Simulate kl coin-flips using the input ω to keep count.
If all the outcomes are "heads", accept.

We argue that this algorithm is a 2PFA for L_m. Consider first the case that $\omega \in L_m$. At each iteration, n=m. The probability of $\mathcal{M}$ rejecting ω in an iteration is

$$P_r = 2^{-k(2n+2)}, \tag{38}$$

and the probability of $\mathcal{M}$ accepting ω in an iteration is

$$P_a = (1 - 2^{-k(2n+2)}) \times 2^{-kl} \geq 2^{-1}2^{-kl} = 2^{-k(2n+1)-1}. \tag{39}$$

Repeating the iteration indefinitely, causes $\mathcal{M}$ to eventually accept ω with probability

$$Pr[\mathcal{M} \textit{ accepts } \omega] = \sum_{i \geq 0} (1 - P_a)^i (1 - P_r)^{i+1} P_a \tag{40}$$

$$= \frac{P_a - P_a P_r}{P_a + P_r - P_a P_r} = \frac{P_a(1 - P_r)}{P_a(1 - P_r) + P_r} \tag{41}$$

$$= \frac{P_a}{P_a + P_r/(1-P_r)} \geq \frac{P_a}{P_a + 2P_r} \tag{42}$$

$$\geq \frac{2^{-k(2n+1)-1}}{2^{-k(2n+1)-1} + 2 \times 2^{-k(2n+2)}} = \frac{1}{1 + 2^{-(k+2)}}. \tag{43}$$

Therefore, the probability that $\mathcal{M}$ accepts before it rejects approaches 1 as k increases.

Consider now the other case that $\omega \notin L_m$. At each iteration, we have $n \neq m$. Therefore, either $2n + 2 \leq l - 1$ or $2m + 2 \leq l - 1$, and whichever case holds, $\mathcal{M}$ will choose the case with probability 1/2. The probability of $\mathcal{M}$ rejecting ω in an iteration is

$$P_r = 2^{-1}2^{-k(2n+2)} + 2^{-1}2^{-k(2m+2)} \geq 2^{-1}2^{-k(l-1)}, \tag{44}$$

and the probability of $\mathcal{M}$ accepting ω in an iteration is

$$P_a = (1 - P_r) \times 2^{-kl} \leq 2^{-kl}. \tag{45}$$

Repeating the iteration indefinitely, causes $\mathcal{M}$ to eventually reject ω with probability

$$Pr[\mathcal{M}\ rejects\ \omega] = \sum_{i \geq 0} (1 - P_a)^i (1 - P_r)^i P_r \tag{46}$$

$$= \frac{P_r}{P_a + P_r - P_a P_r} \geq \frac{P_r}{P_a + P_r} \tag{47}$$

$$\geq \frac{2^{-1}2^{-k(l-1)}}{2^{-kl} + 2^{-1}2^{-k(l-1)}} = \frac{1}{2^{-(k-1)} + 1}. \tag{48}$$

Therefore, the probability that $\mathcal{M}$ rejects before it accepts approaches 1 as k increases.

If we assume that the length of the input $|\omega| = l$, then each iteration takes $\mathbf{O}(l)$ time. The expected number of repeating the algorithm is $2^{\mathbf{O}(l)}$. Hence, the expected running time of $\mathcal{M}$ is $\mathbf{O}(l)2^{\mathbf{O}(l)}$, which is exponential in l. □

Remark 7. *It is easy to show that L_m is non-regular by using the pumping lemma of regular languages.*

In the above theorem, we showed that L_m can be recognized by a 2PFA in exponential expected time $\mathbf{O}(l)2^{\mathbf{O}(l)}$ where l is the length of input. Note that, any 2PFA needs exponential expected time to recognize it, since it is a *non-regular language* [9]. However, we have shown that L_m can be recognized by a 2QCFA in polynomial expected time. Hence, 2QCFA show superiority over 2PFA in recognizing the language L_m.

ACKNOWLEDGMENT

The authors are thankful to the anonymous referees and editor for their comments and suggestions that greatly help to improve the quality of the manuscript.

References

[1] A. Ambainis, M. Beaudry, M. Golovkins, A. Kikusts, M. Mercer, and D. Thénrien, Algebraic Results on Quantum Automata, Theory of Computing Systems 39 (2006), 165-188.

[2] A. Ambainis, R. Freivalds, One-way quantum finite automata: strengths, weaknesses and generalizations, in: Proceedings of the 39th Annual Symposium on Foundations of Computer Science, IEEE Computer Society Press, Palo Alfo, California, USA, 1998, pp. 332-341. Also quant-ph/9802062, 1998.

[3] A. Ambainis, J. Watrous, Two-way finite automata with quantum and classical states, Theoretical Computer Science 287 (2002) 299-311.

[4] A. Brodsky, N. Pippenger, Characterizations of 1-way quantum finite automata, SIAM Journal on Computing 31 (2002) 1456-1478. Also quant-ph/9903014, 1999.

[5] C. Dwork, L. Stockmeyer, A time-complexity gap for two-way probabilistic finite state automata, SIAM J. Comput. 19 (1990) 1011-1023.

[6] C. Dwork, L. Stockmeyer, Finite state verifiers I: The power of interaction, J. ACM 39 (4) (1992) 800-828.

[7] W. Feller, An Introduction to Probability Theory and its Applications, Vol. I, Wiley, New York, 1967.

[8] R. Freivalds, Probabilistic two-way machines, in: Proc. Internat. Symp. on Mathematical Foundations of Computer Science, Strbske Pleso, Lecture Notes in Computer Science, Vol. 188, Springer, Berlin, 1981, pp. 33-45.

[9] A. Greenberg, A. Weiss, A lower bound for probabilistic algorithms for finite state machines. J. Comput. System Sci. 33(1) (1986) 88-105.

[10] L. K. Grover, A fast quantum mechanical algorithm for database search, in: Proceedings of the 28th Annual ACM Symposium on Theory of Computing, Philadelphia, Pennsylvania, USA, 1996, pp. 212-219.

[11] J. Gruska, Quantum Computing, McGraw-Hill, London, 1999.

[12] J. E. Hopcroft, J. D. Ullman, Introduction to Automata Theory, Languages, and Computation, Addision-Wesley, New York, 1979.

[13] A. Kondacs, J. Watrous, On the power of finite state automata, in: Proceedings of the 38th IEEE Annual Symposium on Foundations of Computer Science, 1997, pp. 66-75.

[14] Yu. I. Kuklin, Two-way probabilistic automata, Avtomatika i vyčislitelnaja tekhnika, 1973, No.5, 35-36 (Russian).

[15] L. Z. Li, D. W. Qiu, Determining the equivalence for one-way quantum finite automata, Theoretical Computer Science 403 (2008) 42-51.

[16] L. Z. Li, D. W. Qiu, A note on quantum sequential machines, Theoretical Computer Science 410 (2009) 2529-2535.

[17] C. Moore and J. P. Crutchfield, Quantum automata and quantum grammars, Theoretical Computer Science 237 (2000) 275-306. Also quant-ph/9707031, 1997.

[18] M. A. Nielsen, I. L. Chuang, Quantum Computation and Quantum Information, Cambridge University Press, Cambridge, 2000.

[19] A. Paz, Introduction to Probabilistic Automata, Academic Press, New York, 1971.

[20] D. W. Qiu, Some Observations on Two-Way Finite Automata with Quantum and Classical States, ICIC 2008, LNCS 5226, pp. 1-8, 2008.

[21] D. W. Qiu, L. Z. Li, An overview of quantum computation models: quantum automata, Frontiers of Computer Science in China 2 (2)(2008) 193-207.

[22] D. W. Qiu, S. Yu, Hierarchy and equivalence of multi-letter quantum finite automata, Theoretical Computer Science 410 (2009) 3006-3017.

[23] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM Journal on Computing 26 (5) (1997) 1484-1509.

[24] A. Yakaryilmaz, A. C. C. Say, Efficient probability amplification in twoway quantum finite automata, Theoretical Computer Science, 410 (20) (2009) 1932-1941.

[25] A. Yakaryilmaz, A. C. C. Say, Succinctness of two-way probabilistic and quantum finite automata, Discrete Mathematics and Theoretical Computer Science 12 (4) (2010) 19-40.

[26] A. Yakaryilmaz, A. C. C. Say, Unbounded-error quantum computation with small space bounds, Information and Computation 209 (2011) 873-892.

[27] S. G. Zheng, L. Z. Li, D. W. Qiu, Two-Tape Finite Automata with Quantum and Classical States, International Journal of Theoretical Physics 50 (2011) 1262-1281.